Translated from German

(19) FEDERAL REPUBLIC OF GERMANY

THE GERMAN PATENT AND TRADEMARK OFFICE

(12) **Patent specification**

(10) **DE 199 06 450 C 1**

(51) Int. Cl.^{7}:
H 04 L 9/32
H 04 N 7/16

DE 199 06 450 C 1

(21) File number: 199 06 450.4-31
(22) Date of application: Feb. 16, 1999
(43) Date of disclosure: –
(45) Date of publication of the patent grant: Aug. 17, 2000

Objections may be raised within 3 months from the publication of the patent grant.

(73) Patent holder:
Fraunhofer-Gesellschaft zur Förderung der angewandten Forschung e.V., 80636 Munich, DE

(74) Represented by:
Schoppe, Zimmermann & Stöckeler, 81479 Munich

(72) Inventors:
Rump, Niels, 91056 Erlangen, DE; Koller, Jürgen, 91054 Erlangen, DE; Brandenburg, Karlheinz, Dr., 91054 Erlangen, DE

(56) Documents taken into account for the assessment of the patentability:

DE	196 25 635 C1
US	57 12 914
US	51 59 633

(54) Process and device for generating an encrypted payload stream, and process and device for decrypting an encrypted payload stream

(57) In a process for generating an encrypted payload stream that contains a header and a block with encrypted payload, a payload encryption key is generated for a payload encryption algorithm for the purpose of encrypting payload. The payload is encrypted by means of the generated payload encryption key and the payload encryption algorithm, so as to obtain the block with encrypted payload data of the payload stream. A segment of the payload stream is processed so as to derive information, which characterizes that segment of the payload stream. This information is combined with the payload data by means of an invertible logical operation, so as to obtain a base value. This base value is then encrypted by means of one out of two distinctive encryption keys using an asymmetrical encryption process, the two distinctive encryption keys being the public and, respectively, the private key for the asymmetrical encryption process, so as to obtain an output value, which is an encrypted version of the payload encryption key. Finally, the output value is entered into the header, in order to complete the payload stream. Unauthorized changes to the header or to the payload itself result in the automatic destruction of the payload.

DE 199 06 450 C 1

Description

The present invention refers to the encryption and decryption of payload data, and in particular, to an encryption concept according to which the payload is encrypted by means of a specific encryption key, this key being in turn encrypted so as to achieve a client specific transmission of payload data.

With the development of telecommunication networks, and in particular, as a result of the large-scale expansion of multimedia data compatible personal computers, and recently also of so-called solid-state players, a need has emerged to make digital multimedia data, such as digital audio data and/or digital video data commercially available. The telecommunication networks may be analog telephone lines, digital telephone lines, e.g. ISDN, or the Internet. Commercial suppliers of multimedia products are eager to offer multimedia data for sale or for lease, whereby a customer should, at any time, be able to individually select a specific product from a specific catalog, the utilization of which should of course be restricted to the customer who paid for it.

Contrary to known encrypted TV programs, e.g. programs of the Premiere TV channel, in which the emitted data are equally encrypted for all users who have acquired a suitable decrypting device in exchange for a certain fee, the present invention aims at creating processes and devices enabling an individual, customized and secure encryption and decryption of multimedia data. Unlike the above mentioned TV channels, which provide a fixed program that the user must elect as a whole, the processes and devices of the present invention allow the customer maximum freedom of choice; in other words, the customer has to pay only for those products that he actually wishes to use.

DE 196 25 635 C1 describes processes and devices for encrypting and decrypting multimedia data, whereby the multimedia data are available in the form of an encrypted multimedia data file, which comprises a specification data block and a payload block. Some segments of the specification data block and at least some segments of the payload block are encrypted by means of distinctive keys, whereby in particular symmetrical encryption processes are applied.

On the one hand, symmetrical encryption processes present the advantage of working relatively fast; on the other hand, the user who wishes to decrypt the data file must use the same key as the provider or the supplier, e.g. Deutsche Telekom [German telecommunication company], which encrypted the multimedia data in order to sell them to the customers. Consequently, both the provider and the user, i.e. the customer, use a chart with multiple options of symmetrical encryption algorithms, such as DES or Blowfish, as well as a chart of key options, in such a manner that an entry generated by the provider in the specification data block of the multimedia data is used by the user in order to access his chart of key options, so as to select the proper key for decryption.

The accelerated expansion of the MP3 standard has led to the appearance of so-called solid-state players on the market, used for decrypting and playing back multimedia data. These devices are expected to be very low-priced, and hence should only have a limited amount of storage space and computing power. In contrast to personal computers, in which the available resources exceed by far the resources needed for decrypting multimedia data, solid-state players or HiFi systems or car HiFi devices have to be reasonably priced in order to succeed on the highly competitive market. Hence, for the purpose of decrypting and back-playing the decrypted multimedia data, it is essential to unburden these devices as much as possible in terms of computing power and space available for storage. At the same time, there still remains the requirement that the encryption techniques used be sufficiently secure to guarantee their reliability to the customer and to avoid misuse of encrypted multimedia data. Furthermore, it is imperative to actively counteract copyright violations, in particular when multimedia data are played back without the authorization of the author or of the copyright collecting agency, or when they are altered without authorization.

The US patent No. 5,712,914 presents digital certificates containing multimedia data extensions. According to this patent, two different types of multimedia data are certified. The certification relates to the holder of the rights on the data and to the type of utilization of the data. The certificate is encrypted by means of a public key.

The US patent No. 5,159,633 presents a multimedia network system for transmitting multimedia data, in which both a "secret key system" and a "public key system" are applied. The "secret key" serves to encrypt the payload data, whereas the "public key system" is used to encrypt or, respectively, decrypt any session key or storage type information between the stations.

The present invention endeavors to create an efficient and safe concept for the encryption and, respectively, decryption of multimedia data.

This endeavor is accomplished through a process of generating an encrypted multimedia data stream according to claim 1, a process of decrypting an encrypted multimedia data stream according to claim 17, a device for generating an encrypted multimedia data stream according to claim 27 and a device for decrypting an encrypted multimedia data stream according to claim 28.

The present invention is based on the recognition that, in order to ensure a safe and efficient encryption, a so-called hybrid encryption process must be applied, in which a faster – for example symmetrical – encryption process or scrambling process is applied in order to encrypt or respectively decrypt the payload data themselves, whereas the slower asymmetrical encryption concept is applied only in order to encrypt the payload encryption key for the – for example – symmetrical encryption concept, and to transmit it in this encrypted form to a user, enabling him the decryption of the encrypted payload stream. Furthermore, the encrypted payload stream, which may either be a payload data file or a continuous data stream, should be protected against unauthorized manipulations. In order to achieve this in an effective and optimally economic manner in terms of computing time, the payload stream itself is incorporated in the asymmetrical encryption process for the encryption of the payload encryption key.

It is noteworthy that payload data generally include multimedia data, i.e. audio data, video data or a combination of audio and video data, but also, for example, text data and even binary data, e.g. executable programs. However, in the following presentation, the object of the present invention is described for the sake of convenience on the basis of multimedia data. It is nonetheless evident that all payload data for which encryption is relevant may be processed by means of the devices and the processes according to the invention.

Preferentially, a hash value of a segment of the multimedia data stream is generated. This segment could merely be the header of the multimedia data stream, but it could also include segments of the encrypted or, respectively, unencrypted multimedia data.

An output value in the header, which is supplied to the customer together with the leastwise partly encrypted multimedia data in the form of the multimedia data stream, constitutes, as it were, an encrypted version of the multimedia data encryption key; in order to accurately decrypt this output value so as to obtain the multimedia data encryption key, there may be, apart from the key for the asymmetrical encryption process, also individual data generated by the provider, e.g. license data referring to the manner in which a user is entitled to use the encrypted multimedia data, as well as segments of the multimedia data themselves. Consequently, if a user performs manipulations at the header, e.g. by altering the expiry date of his license to use a specific multimedia section, he can by no means establish the right key for decrypting the encrypted multimedia data, as an accurate decryption of the output value is no longer possible.

Thus, a significant advantage of the process is the fact that, as soon as someone alters the header, the hash value over the header is altered as well. As a result thereof, it is no longer possible to accurately establish the key for decrypting the multimedia data. In other words, any change made to the header automatically leads to the destruction of the multimedia data themselves.

This "implicit" protection of the header does not involve the encryption of the header, and hence the latter needs not be encrypted, which again contributes to saving of resources in the play-back devices. Of course, such encryption of the header would be readily feasible, if it were required.

Similarly, when encrypted or unencrypted multimedia data themselves are incorporated in the encryption of the multimedia data encryption key, a change made to the multimedia data leads to the automatic destruction of all the multimedia data.

Preferential sample embodiments of the present invention are explained in detail below with reference to the attached drawings. These show:

Fig. 1: a multimedia data stream, which can be generated according to the present invention;

Fig. 2: a detailed representation of the header and the payload block of the encrypted multimedia data stream;

Fig. 3: a selection of specific entries in the respective sub-blocks of the header;

Fig. 4: a flow chart representing the process of generating an encrypted multimedia data stream according to the present invention, preferably implemented at the level of the distributor, i.e. the supplier of multimedia data; and

Fig. 5: a process of decrypting an encrypted multimedia data stream according to the present invention, preferably implemented at the level of the customer or user of the multimedia data.

Fig. 1 displays an encrypted multimedia data stream **10**, featuring a header **12** and a payload block **14**, i.e. a block with encrypted multimedia data. The payload block **14** contains encrypted sections **16** and unencrypted sections **18** positioned in between the encrypted sections **16**. Furthermore, a multimedia data stream, which can be generated according to the present invention, contains an additional unencrypted section **20**, positioned immediately after the header 12 and before an encrypted section 16.

Usually, the multimedia data to be encrypted feature some sort of encoding, such as based on an MPEG standard, for example MPEG-2 AAC, MPEG-4 Audio, or MPEG Layer-3. Hence it is sufficient to encrypt certain sections of the multimedia data to be encrypted. This results in a significant reduction of processing costs, both at the level of the provider, where the data are encrypted, and at the level of the customer, who in turn has to decrypt the data. In addition, when the multimedia data are only partly encrypted, the user's satisfaction from hearing or respectively watching, when only the unencrypted multimedia data is used is greatly impaired by the constantly appearing encrypted blocks.

Although **Fig. 1** shows an encrypted multimedia data stream in which the header 12 is positioned at the beginning of the encrypted multimedia data stream, this configuration of the header and the payload block should not refer to the transmission of the encrypted multimedia data stream. The term "header" is only meant to signify that a decryption device, the task of which is decrypting the encrypted multimedia data stream, needs in the first instance at least some segments of the header before the multimedia data themselves can be decrypted. Depending on the transmission medium, the header could also be configured at any location inside the payload block, or could also be received subsequent to specific segments of the payload block, when, for example, a packet-oriented transmission of the multimedia data stream is envisaged, in which different packets, one of which may contain the header and another one may contain a segment of the payload block, are transmitted along different physical transmission paths; thus the reception sequence does by no means have to correspond to the transmission sequence. A decryption device should, however, in such a case, be able to store and to re-arrange the received packets so as to extract information from the header in order to initiate the decryption. Furthermore, the encrypted multimedia data stream could be available in data file form, or else in the form of an actual data stream, for example when a live transmission of a multimedia happening is contemplated. This application will occur in particular in the case of digital user-selective broadcasting.

The length of an encrypted section 16 is represented by the value "amount" 22, whereas the distance in the encrypted multimedia data stream from the beginning of an encrypted section 16 to the beginning of the next encrypted section 16 is referred to as step 24. The length of the next encrypted [according to drawing, this should read "unencrypted" – translator's remark] section 20 is designated by the value "first step" 26.

These values 22, 24 and 26 are, of course, needed to ensure an accurate decryption of the multimedia data in a decryption device, and should therefore be included in the header 12, as explained below.

Fig. 2 provides a detailed representation of the encrypted multimedia data stream 10, which consists of the header 12 and the payload block 14. The header 12 is divided into several sub-blocks, which are explained in detail, in particular in reference to **Fig. 3**. It must be stressed that the number and the function of the sub-blocks may be extended at will. Thus, in **Fig. 2**, only a few sample sub-blocks of the header 12 are represented. The header includes, as shown in **Fig. 2**, a so-called "crypt block" 28 [instead of "29" as per the original – translator's remark], which usually contains relevant information for the encryption of the multimedia data. In addition, the header 12 includes a so-called "license block" 30, which contains data referring to the manner in which a user can or may use the encrypted multimedia data stream. Further, the header 12 includes a payload information block 32, which may contain information regarding the payload block 14 as well as general information about the header 12 itself. The header 12 may further include an "old-header block" 34, which enables a so-called recursive header structure. This block allows the user, who apart from holding a decryption device is also in the possession of an encryption device, to reconstruct (reformat) an encrypted multimedia data stream for other playback appliances in his possession without losing or modifying the original header information supplied by the distributor. Depending on the scope of application, additional sub-blocks may be added to the header 12, for example an IP (=intellectual property) information block according to ISO/IEC 14496-1, MPEG-4 Systems, 1998, containing copyright information.

According to the state of the art, each block may be assigned an internal block structure, which first requires a block identifier, further includes the length of the sub-block, and finally itemizes the block payload itself. Thus, the encrypted multimedia data stream, and in particular the header of the encrypted multimedia data stream, gain increased flexibility in that they allow to respond to new requirements insofar as it is possible to add supplemental sub-blocks or to leave out existing sub-blocks.

Fig. 3 provides an overview of the block payload of the individual sub-blocks displayed in Fig. 2.

We shall first discuss the crypt block 28. It contains an entry for a multimedia data encryption algorithm 40, which identifies the symmetrical encryption algorithm applied in a preferential embodiment, which was applied during the encryption of the multimedia data. The entry 40 could be a chart index, in such a manner that, after reading the entry 40, a decryption device is able to select the same encryption algorithm that was applied by the encryption device out of a large number of encryption algorithms. The crypt block 28 further comprises the "first step" entry 26, the "entry step" 24 and the entry "amount" 22, which were already mentioned in connection with Fig. 1. These entries in the header allow a decryption device to subdivide an encrypted multimedia data stream, thus enabling it to carry out an accurate decryption.

The crypt block 28 further contains an entry for the distributor, provider or supplier 42, which constitutes a code for the distributor who generated the encrypted multimedia data stream. The entry "user" 44 identifies the user, who obtained in whatever way the encrypted multimedia data stream from the distributor identified by entry 42. A possible application of these identifications is to carry out the user identification in a device-specific manner. The entry "user" would then contain the serial number of a PC, a laptop, a car HiFi device, a home HiFi system, etc., authorizing playback only on the specific device. To further enhance the flexibility and/or the protection, instead of using the serial number, which differs from one producer to another but may accidentally be identical, one could use a special identification, for example a logical combination of the hard disc size and the processor's number, etc, e.g. of a PC.

An entry 46 contains an output value that will be discussed in detail further on. Generally speaking, this output value represents an encrypted version of the multimedia data encryption key, which, in conjunction with the multimedia data encryption algorithm identified by the entry 40, is needed in order to accurately decrypt the encrypted multimedia data (sections 16 in Fig. 1) included in the payload block 14. Two further entries are provided in order to achieve sufficient flexibility for future applications: the entry "output value length" 48 and the entry "output value mask" 50. The entry "output value length" 48 specifies the actual length of the output value 46. However, in order to achieve a flexible header format, the number of bytes allocated in the header format to the output value exceeds the actual number of bytes of a current output value. The output value "mask" 50 thus indicates how a shorter output value is distributed, as it were, on a longer output value space. If, for example, the output value length amounts to half the space available for the output value, then the output value mask could be configured in such a manner that the first half of the output value mask is set while the second half is concealed. In this case the output value would simply be entered into the space allocated by the syntax to the header, and would occupy the first half, whereas the second half would be ignored because of the output value "mask" 50.

We shall now discuss the license block 30 of the header 12. The license block includes the entry "bit mask" 52. This entry may contain certain specific information for playback or for the general utilization of the encrypted multimedia data. In particular, this entry could tell a decryption device whether or not the payload may be played back locally. It could also indicate whether the Challenge/Response method – a method enabling efficient data base access, described in the already mentioned German patent DE 196 25 635 C1 – was applied for the purpose of encryption.

The entry "expiry date" 54 specifies the time at which the license to decrypt the encrypted multimedia data stream expires. A decryption device will in this case check the entry "expiry date" 54 and compare it with a built-in time-measuring apliance, so that decryption of the encrypted multimedia data stream is no longer performed if the expiry date has been exceeded. This allows the provider to make encrypted multimedia data available for a limited amount of time, which presents the advantage of a much more flexible handling and pricing. This flexibility is further supported by the entry "starting date" 56, in which it is specified from which point onwards an encrypted multimedia data file may be decrypted. A decryption [instead of "an encryption" as per the original – translator's remark] device will compare the entry "starting date" with its built-in clock, to perform the decryption of the encrypted multimedia data only if the current point in time is later than the starting date 56.

The entry "allowed number of playbacks" 58 indicates how often the encrypted multimedia data stream may be decrypted, i.e. played back. This further contributes to the provider's flexibility, in that he may allow only a certain number of playbacks, for example in relation to a certain price, which is lower than a price that would apply to the unlimited utilization of the encrypted multimedia data stream.

For the purpose of verifying and respectively supporting the entry "allowed number of playbacks" 58, the license block 30 further comprises an entry "actual number of playbacks" 60, which may, for example, be incremented by the value "one" after each decryption of the encrypted multimedia data stream. A decryption device will thus always check whether the entry "actual number of playbacks" is smaller than the entry "allowed number of playbacks". If such is the case, a decryption of the multimedia data is carried out. Otherwise, no more decryptions are carried out.

The entries "allowed number of copies" 62 and "actual number of copies" 64 are implemented in analogy to the entries 58 and 60. Both entries 62 and 64 ensure that a user of the multimedia data copies them only as often as he is allowed to by the provider, or as often as he has paid for when purchasing the multimedia data. The entries 58 to 64 ensure an effective copyright protection, and allow for a selection between private and corporate users, for example by setting the entries "allowed number of playbacks" 58 and "allowed number of copies" 62 at a small value.

The licensing could, for example, be designed so as to allow a certain number of copies (entry 62) of the original, whereas copies made of a copy would not be allowed. The header of a copy would then, unlike the header of the original, have the value "zero" as the entry "allowed number of copies", meaning that this copy is no longer copied by a valid encryption/decryption device.

In the sample multimedia data protection protocol (MMP) displayed here, the header 12 further contains a payload information block 32, which in this case comprises only two block payload entries 66 and 68, the entry 66 containing a hash value over the total header, while the entry 68 identifies the type of hash algorithm used for computing the hash value over the entire header.

In this context, reference is made, for example, to the handbook "Applied Cryptography", Second Edition, John Wiley & Sons, Inc. by Bruce Schneider (ISBN 0 471-11709-9), which contains a detailed description of symmetrical encryption algorithms, asymmetrical encryption algorithms and hash algorithms.

Finally, the header 12 includes the old-header block 34, which, besides the synchronizing information, which is not represented in FIG. 3, comprises the entry "old header" 70. In the entry "old header" 70, the old header may be stored by the provider if a user himself performs an encryption and thus generates a new header 12, so as not to lose essential information entered by the provider into the header. This information may, for example, include copyright information (IP information block), former user information and distributor information, allowing transparent back-tracing of a multimedia data file, which, for example, was decrypted and encrypted several times by different devices, up to the original supplier, the copyright information being stored. It is thus possible to check at any time whether an encrypted multimedia data file was acquired legally or illegally.

Having discussed the format of the encrypted multimedia data stream and the various functionalities of encrypting and decrypting devices, we shall now describe the process according to the invention for the encryption of multimedia data with reference to **Fig. 4**. In a preferential embodiment of the present invention, the encryption process according to the invention is carried out at the distributor's level. The distributor performs preferentially a hybrid encryption process, i.e. a symmetrical encryption process for the encryption of the multimedia data and an asymmetrical encryption process for the encryption of the multimedia data encryption key.

A customer or user, who wishes to acquire multimedia data from a distributor, first contacts the distributor, and may, for example, provide him the number of his credit card, which the distributor charges for payments due. The customer then receives from the distributor a chart of symmetrical encryption processes. In addition, the distributor and the customer exchange their respective public keys. When the user now orders a specific multimedia product from the distributor, the latter performs a customized encryption for that customer.

The detailed steps for generating the encrypted multimedia data stream may be as follows:

The distributor first creates the header 12 for the multimedia data file, as far as possible at this point in time (100). As displayed in Fig. 3, at this point the output value is not yet available. Thus, in step 100, in which the header 12 is created as far as possible, the entry for the output value is released. At that stage, however, all other entries into the crypt block and all other entries into the license block are already in place. The hash value or, respectively, the digital signature in the entry 66 for the entire header does not yet exist, and therefore this entry too is still free. The entry "old-header" 70 will also, most probably, remain free when the multimedia data file is encrypted for the first time by the distributor. However, if the distributor acquired the encrypted multimedia data file from another distributor, then the entry 70 might already be filled in. In step 102, the distributor generates a multimedia data encryption key, which, together with the multimedia data encryption algorithm identified by entry **40** (**Fig. 3**), allows the encryption of the multimedia data, which is carried out in step **104**.

According to the present invention, a hash value is computed over the header, whereby specific segments have a pre-defined value (step **106**). The detailed representation of the header in **Fig. 3** features a column **107** at the right-hand side, indicating which segments or entries in the header **12** are given a pre-defined value when the hash value is computed during step **106** (**Fig. 4**). A pre-defined value is given, in particular, to the entry "output value" **46**, the entry "actual number of playbacks" **60**, the entry "actual number of copies" **64** and the entry "hash value over the header" **66**, and also, possibly, the entry "old header" **70**, as indicated by the dotted-line cross for the entry **70**. Specific segments of the header must be allocated a pre-defined value when the hash value is computed in step **106**, as they are not yet fixed (output value **46**) or as they are modified by a decryption device (entries **60** and **64**). Entry **66**, i.e. the hash value over the header, is not yet fixed either, as it is to include, of course, the output value **46**.

The entries "distributor" **42** and "user" **44** as well as the entries in the license block **30** are incorporated in the computation of the hash value during step **106** (**Fig. 4**); thus personalization, or respectively, protection of the license block entries is already achieved, as the hash value computed during step **106** is combined with the multimedia data encryption key so as to obtain a base value (step **108**).

Subsequently, the base value obtained during step **108** is asymmetrically encrypted by means of the customer's public key (step **110**). In order to put the encrypted multimedia data stream in a transferable format, the header is then completed (step **112**) in that the output value **46** is entered into the header created in step **100**.

It is possible to deviate from the embodiment represented in **Fig. 4** by changing the sequence of the steps. For example, it is possible to first carry out the full encryption of the multimedia data encryption key and only then to encrypt the multimedia data stream. Another possibility is to compute the hash value over the header before generating the multimedia data encryption key. Additional variants are also feasible. Obviously, step **108** can only be carried out after the hash value is computed. Furthermore, step **110** can only be carried out when the base value is available.

For the encryption of the multimedia data by means of the multimedia data encryption key in step **104**, a symmetrical encryption process is preferable, as in this instance there may be relatively a lot of data to be encrypted or, respectively, decrypted. It is a well known fact that symmetrical encryption processes work faster than asymmetrical encryption processes, such as are applied in step **110** for the encryption of the multimedia data encryption key.

It is also deemed preferential to generate the multimedia data encryption key by means of a random number generator, so that the base value for one and the same customer, generated during step **108**, receives each time a different form, so as to make it as difficult as possible for an invader to attack the cryptographic system.

The operation used for combining the hash value with the multimedia data encryption key should, as clarified in relation to **Fig. 5**, be an auto-inverse operation. One such auto-inverse operation may be the XOR operation (exclusive OR). The meaning of "auto-inverse" is that when such an operation is performed twice, its result equals the initial value. Moreover, it is possible that the combination function of **Fig. 5** is the inverse function of the combination function of **Fig. 4**. Hence, the combination function must necessarily be invertible, i.e. there should be an inverse function to that function.

In step **110**, according to the present invention, an asymmetrical encryption process is carried out. It is a well known fact that in an asymmetrical encryption process, there are two encryption keys, which can both perform the encryption or, respectively, the decryption, and yet differ from one another. One of the keys is designated as the "private key" and the other one as the "public key". Normally, asymmetrical encryption processes are characterized in that data to be encrypted, which have been encrypted by means of the private key, may be decrypted by means of the public key. Analogically, data to be encrypted, which have been encrypted by means of the public key, are decrypted by means of the private key. This implies that the private and the public keys are in principle mutually interchangeable.

One aspect of the present invention consists in that the header is incorporated via steps **106** and **108** in the encryption of the multimedia data encryption key. Alternatively, however, it is also possible to incorporate segments of the multimedia data encryption key, whereby unauthorized manipulations of the payload data would render the entire multimedia data stream inoperative, as it would no longer be possible to calculate the multimedia data encryption key in the decryption process.

Although it is mentioned in step 106 that a hash value is computed over the header, it should be noted that it is possible to perform any processing of a segment of the multimedia data stream in order to derive information which characterizes that segment of the multimedia data stream. The more elaborate the hash algorithm applied, the better the encrypted multimedia data stream is protected against invaders who may wish to crack it, e.g. in order to modify the license information or, respectively, the distributor or user information for their (unauthorized) purposes, .

In the following section, reference is made to Fig. 5, which displays a flow chart of a decryption process as may be applied by a customer. In step 120, the customer first reads the output value from the header of the encrypted multimedia data stream. He then performs a decryption of this output value by means of the appropriate asymmetrical decryption process (step 122). Thereupon the customer's decryption device computes a hash value over the header, whereby the specific segments, which were given a pre-determined value during encryption, are given the same pre-determined value during step 124. The hash value is then combined with the decrypted output value (step 122), so as to obtain the multimedia data encryption key (step 126). Finally, during step 128, the encrypted multimedia data are decrypted by means of the multimedia data encryption key obtained during step 126.

It is evident that the decryption process in fact represents the reversal of the encryption process as described by the flow chart in **Fig. 4**. Of course, also in the decryption process shown in **Fig. 5**, it is possible to switch between some of the steps. For example, it is possible to first compute the hash value over the header (step 124) and then to decrypt the output value by means of the public key (step 122). Reading the output value from the header (step 120) may also be performed, for example, only after step 124, though by all means before step 126. Furthermore, step 128 is possible only after step 126 has been completed, as the latter provides the multimedia data encryption key.

The decryption process displayed in **Fig. 5** evidences, through step 124, what happens when a customer has altered the header 12, which is normally unencrypted and which is readily accessible to invasion. An alteration of the license information, such as the starting date and the expiry date, would necessarily cause the hash value over the header, computed during step 124, to differ from the hash value computed during step 106 (Fig. 4) at the time of the encryption. The new computation of the hash value during step 126 (Fig. 5) will thus no longer lead to the accurate multimedia data encryption key, as both hash values, i.e. the hash value during encryption and the hash value during decryption, differ from one another. Hence the multimedia data as a whole are inoperative, as they can no longer be accurately decrypted, because it is no longer possible to calculate the multimedia data encryption key applied by the encryption device, due to the manipulation of the header. Thus, any change to the header automatically entails the destruction of the multimedia data themselves.

Patent claims

1. Process for generating a payload stream (10) containing a header (12) and a payload block (14) with encrypted payload data, which comprises the following steps:

Generating (102) a payload encryption key for a payload encryption algorithm for the encryption of payload data;

Encrypting (104) payload data by means of the payload encryption key and the payload encryption algorithm, so as to obtain an encrypted section (16) of the payload block (14) of the payload stream (10);

Processing (106) a segment of the payload stream (10) in order to derive information characterizing that segment of the payload stream;

Combining (108) the information with the payload encryption key by means of an invertible logical operation, so as to obtain a base value;

Encrypting (110) the base value by means of one out of two distinctive encryption keys through an asymmetrical encryption process, the two distinctive encryption keys being the public and, respectively, the private key for the asymmetrical encryption process, so as to obtain an output value (46), which is an encrypted version of the payload encryption key; and

Entering (112) the output value (46) into the header (12) of the payload stream (10).

2. Process according to claim 1, in which the payload encryption algorithm is a symmetrical encryption algorithm.

3. Process according to claim 1 or 2, in which the invertible logical operation is an auto-inverse operation comprising an exclusive OR operation.

4. Process according to one of the above claims, in which one of the two distinctive encryption keys is the private key of a generator of the payload stream

or the public key of a consumer of the payload stream.

5. Process according to one of the above claims, in which the segment of the payload stream that is processed (106) so as to derive the information comprises at least a segment of the header (12).

6. Process according to one of the above claims, in which the processing step (106) comprises the computation of a hash value.

7. Process according to one of the above claims, which, furthermore, includes the following step:

Identifying the algorithm applied during the processing step (106) by means of an entry (68) in the header.

8. Process according to one of the above claims, which furthermore includes the following step:

Entering license data (30) into the header (12), these data referring to the manner in which the payload stream (10) may be used.

9. Process according to claim 8, in which the license data (30) indicate how often the payload stream may be played back (58), and how often it was already played back (60).

10. Process according to claim 8 or 9, in which the license data (30) indicate how often the content of the payload stream may be copied (62), and how often is was already copied (64).

11. Process according to one of the claims 8 through 10, in which the license data (30) indicate as of what date the payload stream may no longer be used (54).

12. Process according to one of the claims 8 through 11, in which the license data (30) indicate as of what date the payload stream may be decrypted (56).

13. Process according to one of the claims 8 through 12, in which the segment of the payload stream that is processed so as to derive the information (106) contains the license data (30).

14. Process according to one of the above claims, in which the processing step also comprises the following sub-steps:

Setting the entry (46) for the output value in the header (12) at a defined value; processing (106) the entire header, including the entry that was set at a defined value (46).

15. Process according to one of the above claims, which, furthermore, comprises the following steps:

Identification of the distributor (42) of the payload stream through a distributor entry (42) in the header (12);

Identification of the user (44) of the payload stream through a user entry (44) in the header (12) of the payload stream,

whereby the distributor entry (42) and the user entry (44) belong to that segment of the payload stream (10), which is processed (106) so as to derive the information.

16. Process according to one of the above claims, which, furthermore, comprises the following step:

Identification of the payload encryption algorithm through an entry (40) in the header (12) of the payload stream (10).

17. Process for decrypting an encrypted payload stream (10), which comprises a header (12) and a payload block (14) with encrypted payload data, whereby the header (12) contains an output value (46), generated by encryption of a base value through an asymmetrical encryption process and by means of one out of two distinctive encryption keys, one of which is a private key and the other a public key, whereby the base value constitutes a combination of a payload encryption key, by which the encrypted payload is encrypted through a payload encryption algorithm, with information that is derived through a specific processing and that unambiguously characterizes a specific segment of the payload stream (10); this process comprises the following steps:

Obtaining (120) the output value (46) from the header (12);

Decrypting (122) the output value (46) by means of the other key [and] of the asymmetrical encryption process, so as to obtain the base value;

Processing (124) a segment of the payload stream (10) through the processing method used during the encryption for the purpose of deriving information characterizing that segment, whereby that segment corresponds to the specific segment [processed] during encryption;

Combining (126) the information with the base value by means of the corresponding operation, as it was applied during the encryption, so as to obtain the payload encryption key; and

Decrypting (128) the block (14) containing encrypted payload data by means of the payload encryption key and the payload encryption algorithm applied during the encryption.

18. Process according to claim 17, in which the header (12) contains license information (30) indicating in which manner the payload stream (10) may be used.

19. Process according to claim 17 or 18, in which the segment processed in order to derive the information is the header (12).

20. Process according to claim 18 or 19, which furthermore comprises the following steps:

Checking whether the license information (30) allows a decryption; and,

in the event that a decryption is not allowed, abort the decryption process.

21. Process according to one of the claims 17 through 20, in which the header (12) contains a user entry (44), and which furthermore comprises the following steps:

Checking, by means of the user entry (44), whether a current user is authorized; and,

in the event that the user is not authorized, abort the decryption process.

22. Process according to one of the claims 17 through 21, in which the one key that was used during the encryption is the private key of the asymmetrical encryption process, whereas the other key, which is used during the decryption, is the public key of the asymmetrical encryption process.

23. Process according to one of the claims 17 through 21, in which the one key that was used during the encryption is the public key of the asymmetrical encryption process, whereas the other key, which is used during the decryption, is the private key of the asymmetrical encryption process.

24. Process according to one of the claims 17 through 23, in which the processing step (124) comprises the computing of a hash value.

25. Process according to one of the claims 17 through 24, in which a segment of the header (12), which was set at a defined value for the processing step during encryption, is set at the same defined value for the processing step (124) during decryption.

26. Process according to claim 25, in which the segment of the header (12) which is set at a defined value contains the entry for the output value (46) of the header (12).

27. Process according to one of the claims 17 through 26, in which the combination step (126) includes the use of an exclusive OR operation.

28. Device for generating an encrypted payload stream, which comprises a header (12) and a payload block (14) with encrypted payload data, with the following characteristics:

A device for generating (102) a payload encryption key for a payload encryption algorithm for the encryption of payload data;

A device for encrypting (104) payload data by means of the payload encryption key and the payload encryption algorithm, in order to obtain an encrypted section (16) of the payload block (14) of the payload stream (10);

A device for processing (106) a segment of the payload stream (10), in order to derive information characterizing that segment of the payload stream;

A device for combining (108) the information with the payload encryption key by means of an invertible logical operation, in order to obtain a base value;

A device for encrypting (110) the base value by means of one out of two distinctive encryption keys through an asymmetrical encryption process, the two distinctive keys being the public and, respectively, the private key for the asymmetrical encryption process, so as to obtain an output value (46), which is an encrypted version of the payload encryption key; and

A device for entering (112) the output value (46) into the header (12) of the payload stream (10).

29. Device for decrypting an encrypted payload stream (10), which contains a header (12) and a block (14) with encrypted payload data, whereby the header (12) contains an output value (46), generated by encryption of a base value through an asymmetrical encryption process and by means of one out of two distinctive encryption keys, one of which is a private key and the other a public key, whereby the base value constitutes a combination of a payload encryption key, by which the encrypted payload data are encrypted through a payload encryption algorithm, with information that is derived through a specific processing and that unambiguously characterizes a specific segment of the payload stream (10); this device features the following characteristics:

A device for obtaining (120) the output value (46) from the header (12);

A device for decrypting (122) the output value (46) by means of the other key and of the asymmetrical encryption process, so as to obtain the base value;

A device for processing (124) a segment of the payload stream (10) through the processing method used during the encryption for the purpose of deriving information characterizing that segment, whereby that segment corresponds to the specific segment [processed] during encryption;

A device for combining (126) the information with the base value by means of the corresponding operation, as it was applied during the encryption, so as to obtain the payload encryption key; and

A device for decrypting (128) the block (14) containing encrypted payload data by means of the payload encryption key and the payload encryption algorithm applied during the encryption.

30. Device according to claim 28 or 29, implemented as personal computer, as HiFi system, as car HiFi device, as solid-state player or as playback device with hard disc or CD-ROM.

See drawings on 4 attached pages

Fig. 1

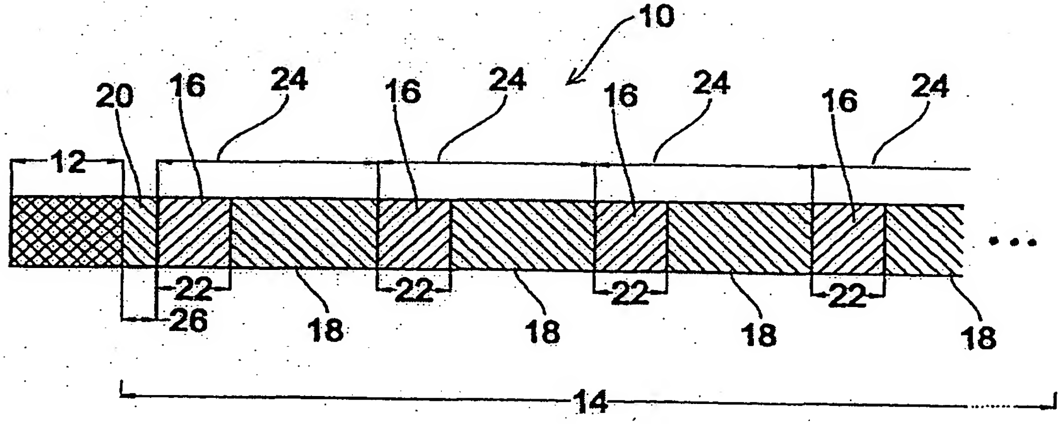

10
20
16
24
16
24
16
24
16
24
12
22
22
22
22
26
18
18
18
18
14

Fig. 2

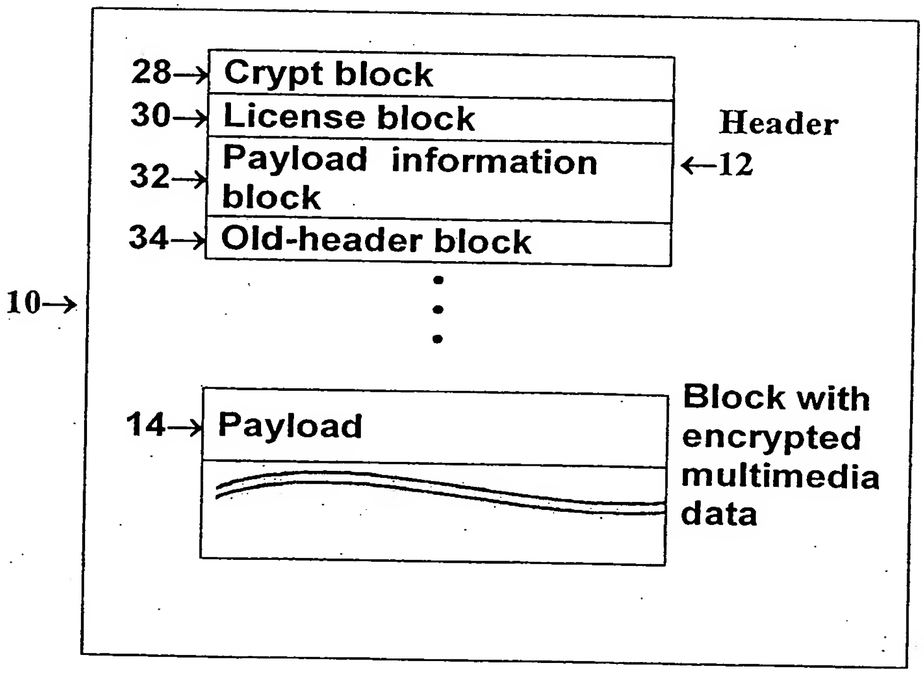

28
Crypt block
30
License block
32
Payload information block
34
Old-header block
Header
12
10
14
Payload
Block with encrypted multimedia data

Fig. 3

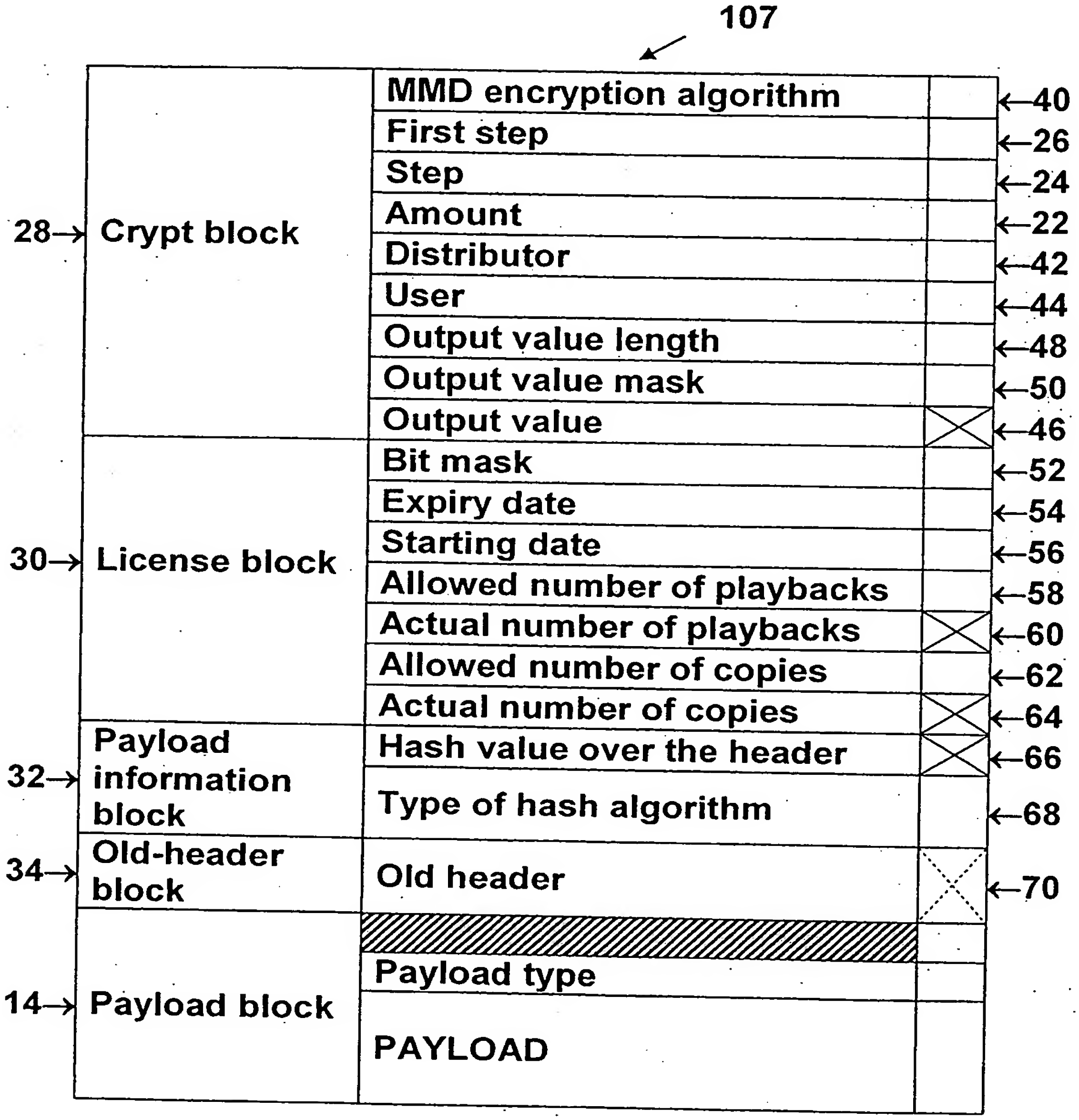

107
28
Crypt block
MMD encryption algorithm
40
First step
26
Step
24
Amount
22
Distributor
42
User
44
Output value length
48
Output value mask
50
Output value
46
30
License block
Bit mask
52
Expiry date
54
Starting date
56
Allowed number of playbacks
58
Actual number of playbacks
60
Allowed number of copies
62
Actual number of copies
64
32
Payload information block
Hash value over the header
66
Type of hash algorithm
68
34
Old-header block
Old header
70
14
Payload block
Payload type
PAYLOAD

Fig. 4

AT THE DISTRIBUTOR'S LEVEL

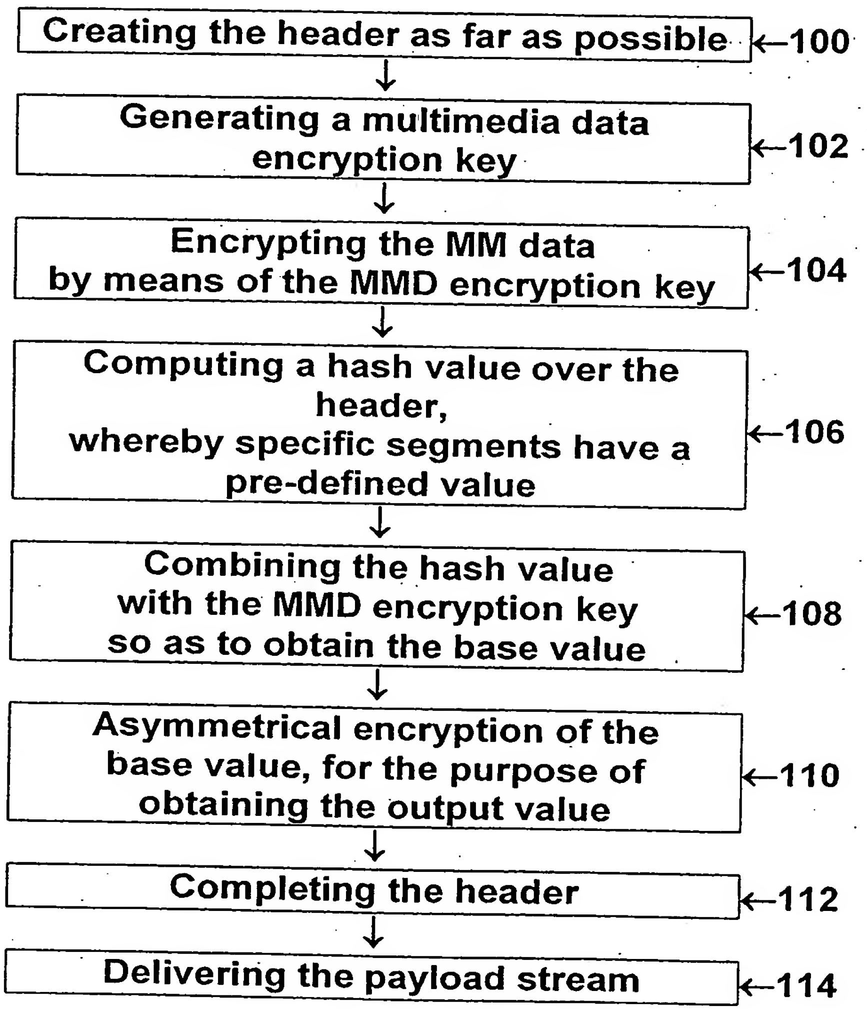
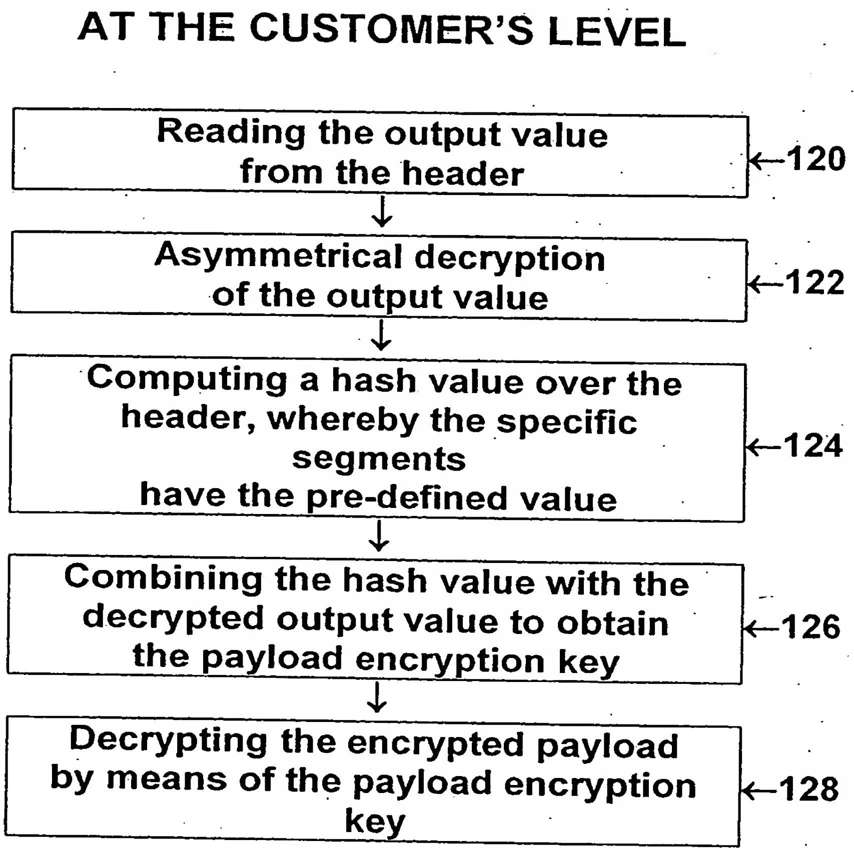

Creating the header as far as possible
100
Generating a multimedia data encryption key
102
Encrypting the MM data by means of the MMD encryption key
104
Computing a hash value over the header, whereby specific segments have a pre-defined value
106
Combining the hash value with the MMD encryption key so as to obtain the base value
108
Asymmetrical encryption of the base value, for the purpose of obtaining the output value
110
Completing the header
112
Delivering the payload stream
114

Fig. 5


AT THE CUSTOMER'S LEVEL
Reading the output value from the header
120
Asymmetrical decryption of the output value
122
Computing a hash value over the header, whereby the specific segments have the pre-defined value
124
Combining the hash value with the decrypted output value to obtain the payload encryption key
126
Decrypting the encrypted payload by means of the payload encryption key
128